# Ch3 - COMBINATIONAL LOGIC CIRCUITS AND BOOLEAN ALGEBRA

Digital Electronics (Universiti Teknologi MARA)

# CHAPTER 3

# COMBINATIONAL LOGIC CIRCUITS

# AND

# BOOLEAN ALGEBRA

**Objectives**

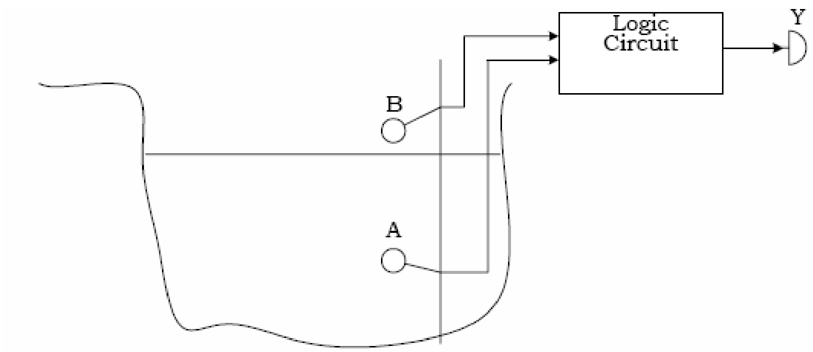
- **Designing Logic Circuit**
- **Deriving Sum of Product and Product of Sum Expression**
- **Boolean Algebra and DeMorgan's Theorem**
- **Simplifying Boolean Algebra**
- **Universality of NAND and NOR gates.**

## 3.1 Designing a Combinational Logic Circuits

The purpose of studying combinational logic circuits is to design circuits to perform specific logic functions as required by applications. The circuits are constructed by connecting together logic gates. To design a logic circuit, the application must be defined in a truth table.

Suppose we would like to design a logic circuit to monitor the water level at a river bank. The buzzer Y will be activated if the water level is below level A or above the level B. There are three steps in designing a combinational logic circuit.

1. Derive the truth table.
2. Derive the simplified Boolean expressions.
3. Draw the circuit.




The first step is to set up the truth table based on the problem statement. A truth table contains all possible input combinations and the output for each input combination. The circuit has 2 inputs, float switches A and B and a buzzer Y as its output. The logic levels to represent the input and output:

Input
Float off = 0
Float on = 1

Output
Buzzer on = 1
Buzzer off = 0

There are four input combinations: 00, 01, 10 and 11.

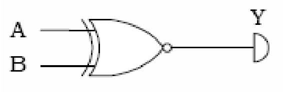
$AB = 00$ The water level is below A. Y=1
$AB = 10$ The water level is between A and B. Y=0
$AB = 11$ The water level is above B. Y=1
$AB = 01$ The float B is not functioning. Y=0

The truth table:

| Float | | Buzzer |
|---|---|---|
| A | B | Y |
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

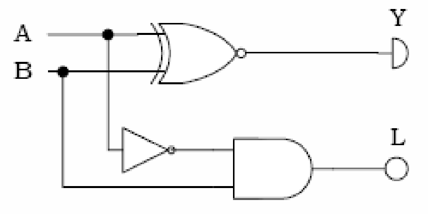
From the truth table, the Boolean expression for Y can be written as:
$Y = \overline{A}\,\overline{B} + AB = \overline{A \oplus B}$

The logic circuit:




Suppose we add another output L to warn that the float B is not functioning. The truth table has 2 outputs as shown below. The Boolean expression is $L = \overline{A}B$.

| Float | | Buzzer | Red Light |
|---|---|---|---|
| A | B | Y | L |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 |




In the next section, we will learn how derive a Boolean expression from the truth table.

## 3.2 Deriving a Boolean Expression from a Truth Table

From a truth table, we can write two forms of logic or Boolean expressions: Sum of Product (SOP) and Product of Sum (POS) expressions.

**Sum of Product** (SOP) expression is the sum of all **product terms** when the output is a logic 1. What is a product term? A product term is formed by ANDing the complemented or uncomplemented input variables. For a 2 variables truth table, the product term for AB=00 is $\overline{A} \cdot \overline{B}$ , AB=01 is $\overline{A} \cdot B$, AB=10 is $A \cdot \overline{B}$ and AB=11 is $AB$.

$00 \rightarrow \overline{A} \cdot \overline{B}$
$01 \rightarrow \overline{A} \cdot B$
$10 \rightarrow A \cdot \overline{B}$
$11 \rightarrow AB$

Product Terms

Let us derive the SOP expression from the OR gate truth table. There are three outputs where the output Y=1. The SOP expression is $Y = \overline{A}B + A\overline{B} + AB$.

| Product term | A | B | Y |
|---|---|---|---|
| $\overline{A}\,\overline{B}$ → | 0 | 0 | 0 |
| $\overline{A}B$ → | 0 | 1 | 1 |
| $A\overline{B}$ → | 1 | 0 | 1 |
| $AB$ → | 1 | 1 | 1 |

OR gate Truth Table

The SOP expression from the truth table in part 3.1 can be written as $Y = \overline{A} \cdot \overline{B} + AB$.

**Product of Sum** (POS) expression is the product all **sum terms** when the output is a logic 0. What is a sum term? A sum term is formed by ORing the complemented or uncomplemented input variables. For a 2 variables truth table, the sum term for AB=00 is $A + B$, AB=01 is $A + \overline{B}$, AB=10 is $\overline{A} + B$ and AB=11 is $\overline{A} + \overline{B}$.

$00 \rightarrow A + B$
$01 \rightarrow A + \overline{B}$
$10 \rightarrow \overline{A} + B$
$11 \rightarrow \overline{A} + \overline{B}$

Sum terms

The OR gate truth table contains only one output where the output Y=0. The POS expression can be written as $Y = A + B$.

| Sebutan Jumlah | A | B | Y |
|---|---|---|---|
| $A + B$ → | 0 | 0 | 0 |
| $A + \overline{B}$ → | 0 | 1 | 1 |
| $\overline{A} + B$ → | 1 | 0 | 1 |
| $\overline{A} + \overline{B}$ → | 1 | 1 | 1 |

So, the Boolean expression for an OR gate can be written in two forms:

SOP expression: $Y = \overline{A}B + A\overline{B} + AB$
POS expression: $Y = A + B$

The two forms of Boolean expression for the truth table in section 3.1 can be written as follows:

SOP expression: $Y = \overline{A} \cdot \overline{B} + AB$
POS expression: $Y = \left(A + \overline{B}\right)(\overline{A} + B)$

In the next section, we will prove that both expressions are equivalent. The expression $Y = \overline{A}B + A\overline{B} + AB$ can be simply as $Y = A + B$ using Boolean algebra.

## 3.3 Boolean Algebra

The original mathematical theory on Boolean algebra was first introduced by George Boole (1815-1864), an English mathematics professor in 1845. The basic laws of Boolean algebra are mathematically equivalent to the rules of standard variable algebra. As in standard algebra, there are three laws for Boolean algebra: Communicative, Associative and Distributive Laws.

**1. Communicative Law**

- $\mathrm{A + B = B + A}$
- $\mathrm{AB = BA}$

Example:
$\overline{\mathrm{A + B}} = \overline{\mathrm{B + A}}$
$\overline{\mathrm{AB}} = \overline{\mathrm{BA}}$
$\mathrm{A \oplus B = B \oplus A}$
$\overline{\mathrm{A \oplus B}} = \overline{\mathrm{B \oplus A}}$
$\mathrm{A + B + C = B + C + A = C + A + B}$

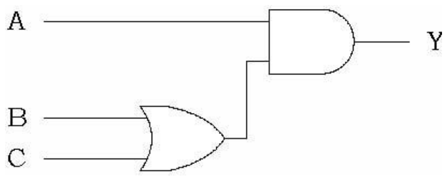
**2. Associative Law**

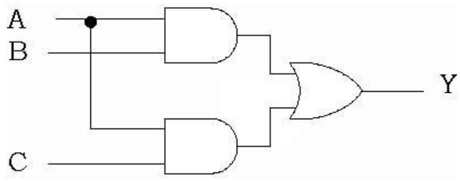
- $\mathrm{A + (B + C) = (A + B) + C}$
- $\mathrm{A(BC) = (AB)C}$

**3. Distributive Law**

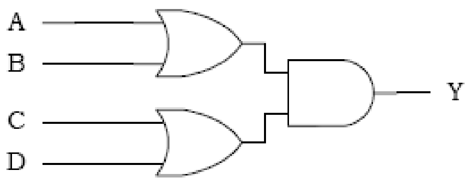
- $\mathrm{A(B + C) = AB + AC}$
- $\mathrm{(A + B)(C + D) = AC + AD + BC + BD}$

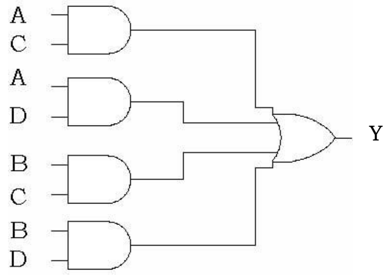
The circuit for $Y = A(B + C)$ uses 2 gates whereas $Y = AB + AC$ uses 3 gates.




The circuit for $Y = (A + B)(C + D)$ uses 3 gates, whereas $Y = AC + AD + BC + BD$ uses 5 gates.




In addition to the three basic laws, there are several Boolean theorems that are very useful in simplifying Boolean expressions and logic circuits.

**Boolean algebra theorems**

**Theorem 1**: **A.0=0**
Example:
- AB.0 = 0
- (A+B)(C+D).0 = 0

**Theorem 2**: **A.1=A**
Example:
- AB.1 = AB
- (A+B)(C+D).1 = (A+B)(C+D)

**Theorem 3**: **A+0=A**
Example:
- AB + 0 = AB
- A + B + 0 = A + B

**Theorem 4**: **A+1=1**
Example:
- AB + 1 = 1
- A+B+C +1 = 1
- (A+B)(C+D) + 1 = 1

**Theorem 5**: **A.A=A**

Example:

- (AB).(AB) = AB
- (A+B)(A+B) = A+B
- A.B.A = A.B
- A.B.C.A.B.C.A = A.B.C

**Theorem 6**: **A+A=A**

Example:

- AB + AB = AB
- A+A+B+C = A+B+C

**Theorem 7**: $A.\overline{A} = 0$

Example:

- $AB.\overline{AB} = 0$
- $(A + B)(\overline{A + B}) = 0$
- $A.B.\overline{A} = 0.B = 0$

**Theorem 8**: $A + \overline{A} = 1$

Example:

- $AB + \overline{AB} = 1$
- $(A + B) + (\overline{A + B}) = 1$
- $A + B + \overline{A} = 1 + B = 1$

**Theorem 9**: $\overline{\overline{A}} = A$

Example:

- $\overline{\overline{AB}} = AB$
- $\left(\overline{\overline{A + B}}\right) = A + B$

**Theorem 10**: **A+BC = (A+B)(A+C)**

<u>Example 1:</u>

$$\begin{aligned} A + BCD &= (A + B)(A + CD) \\ &= (A + B)(A + C)(A + D) \end{aligned}$$

<u>Example 2:</u>

$$\begin{aligned} A + B + CD &= (A + B) + CD \\ &= (A + B + C)(A + B + D) \end{aligned}$$

<u>Example 3:</u>

$$\begin{aligned} A + \overline{A}.B &= (A + \overline{A})(A + B) \\ &= 1(A + B) \\ &= A + B \end{aligned}$$

**Theorem 11**: **A+AB = A**

**Proof:**

$$\begin{aligned} A + AB &= A(1 + B) \\ &= A.1 \\ &= A \end{aligned}$$

Example 1:

$$\begin{aligned} A + ABC &= A(1 + BC) \\ &= A.1 \\ &= A \end{aligned}$$

Example 2:

$$\begin{aligned} AB + ABC &= AB(1 + C) \\ &= AB.1 \\ &= AB \end{aligned}$$

## 3.4 Simplifying Expressions using Boolean Algebra

In this section we will learn how to use Boolean algebra to simplify Boolean expressions.

**Example 1:**

Simplify $Y = \overline{A} + ABC$

$$\begin{aligned} Y &= \overline{A} + ABC \\ &= (\overline{A} + A)(\overline{A} + BC) \\ &= 1(\overline{A} + BC) \\ &= \overline{A} + BC \end{aligned}$$

**Example 2:**

Simplify $Y = A + \overline{A}.B + \overline{A}.B.C$

Solution 1:

$$\begin{aligned} Y &= A + \overline{A}.B + \overline{A}.B.C \\ &= A + \overline{A}.B(1 + C) \\ &= A + \overline{A}.B \\ &= A + B \end{aligned}$$

Solution 2:

$$\begin{aligned} Y &= A + \overline{A}.B + \overline{A}.B.C \\ &= A + B + \overline{A}.B.C \\ &= A + B + B.C \\ &= A + B(1 + C) \\ &= A + B \end{aligned}$$

**Example 3:**
Prove that the SOP and POS expressions for an OR gate are equivalent.
SOP expression $Y = \overline{A}.B + A.\overline{B} + AB$
POS expression $Y = A + B$

$$\begin{aligned} Y &= \overline{A}.B + A.\overline{B} + AB \\ &= \overline{A}.B + A(\overline{B} + B) \\ &= \overline{A}.B + A \\ &= (A + \overline{A})(A + B) \\ &= (A + B) \end{aligned}$$

**Example 4:**
Prove that the SOP and POS expressions for an AND gate are equivalent.
POS expression $Y = (A + B)(A + \overline{B})(\overline{A} + B)$
SOP expression $Y = A.B$

$$\begin{aligned} Y &= (A + B)(A + \overline{B})(\overline{A} + B) \\ &= \left(AA + A.\overline{B} + AB + B.\overline{B}\right)\left(\overline{A} + B\right) \quad \textit{Expand the expression} \\ &= \left(A + A.\overline{B} + AB + 0\right)\left(\overline{A} + B\right) \\ &= \left(A + A.\overline{B} + AB\right)\left(\overline{A} + B\right) \\ &= A\left(1 + \overline{B} + B\right)\left(\overline{A} + B\right) \quad \textit{Factorise } A \\ &= A(1)\left(\overline{A} + B\right) \\ &= A\left(\overline{A} + B\right) \\ &= A.\overline{A} + AB \\ &= 0 + AB \\ &= AB \end{aligned}$$

**Example 5:**
Prove that $\overline{A}.\overline{B}.\overline{C} + \overline{A}.\overline{B}.C + A.\overline{B}.\overline{C} = \overline{A}.\overline{B} + \overline{B}.\overline{C}$

$$\begin{aligned} &\overline{A}.\overline{B}.\overline{C} + \overline{A}.\overline{B}.C + A.\overline{B}.\overline{C} \\ &= \overline{A}.\overline{B}\left(\overline{C} + C\right) + A.\overline{B}.\overline{C} \quad \textit{Factorise } \overline{A}.\overline{B} \\ &= \overline{A}.\overline{B}(1) + A.\overline{B}.\overline{C} \\ &= \overline{A}.\overline{B} + A.\overline{B}.\overline{C} \\ &= \overline{B}\left(\overline{A} + A.\overline{C}\right) \quad \textit{Factorise } \overline{B} \\ &= \overline{B}\left(\overline{A} + A\right)\left(\overline{A} + \overline{C}\right) \quad \textit{Use theorem } 10 \rightarrow A + BC = (A + B)(A + C) \\ &= \overline{B}\left(\overline{A} + \overline{C}\right) \\ &= \overline{A}.\overline{B} + \overline{B}.\overline{C} \end{aligned}$$

## 3.5 Theorem DeMorgan

DeMorgan's theorems was first proposed by an English mathematician Augustus DeMorgan (1806-1871) and later expanded on by George Boole. DeMorgan's theorem can be used to simplify expressions in which a product or sum of variables is inverted. The two theorems are:

$$\overline{A.B} = \overline{A} + \overline{B}$$
$$\overline{A+B} = \overline{A}.\overline{B}$$

The first theorem says, when the AND product of two variables is inverted $\overline{A.B}$, this is the same as inverting each variable individually and then ORing these inverted variables $\overline{A} + \overline{B}$.

The second theorem says, when the OR sum of two variables $\overline{A+B}$ is inverted, this is the same as inverting each variable individually and then ANDing these inverted variables $\overline{A}.\overline{B}$.

From the first theorem, a NAND gate is equivalent to an AND gate with inverters on each input. A NOR gate is equivalent to an OR gate with inverters on each input as illustrated below:

$\overline{A.B} = \overline{A} + \overline{B}$


$\overline{A+B} = \overline{A}.\overline{B}$


These theorems can be expanded to two or more variables.

$$\overline{A.B.C} = \overline{A} + \overline{B} + \overline{C}$$
$$\overline{A+B+C} = \overline{A}.\overline{B}.\overline{C}$$

Although these theorems have been stated in terms of single variables A and B, they are equally valid for expressions that contain more than one variable.

<u>Example:</u>

- $\overline{(A + B).C} = \overline{A + B} + \overline{C} = \overline{A}.\overline{B} + \overline{C}$
- $\overline{\overline{A}.B} = \overline{\overline{A}} + \overline{B} = A + \overline{B}$
- $\overline{\overline{\overline{A}.B}.C} = \overline{\overline{\overline{A}.B}} + \overline{C} = \overline{A}.B + \overline{C}$
- $\overline{AB + C} = \overline{AB}.\overline{C} = (\overline{A} + \overline{B}).\overline{C}$
- $\overline{A(B + C) + D} = \overline{AB + AC + D} = \overline{AB}.\overline{AC}.\overline{D} = (\overline{A} + \overline{B})(\overline{A} + \overline{C})\overline{D}$
- $\overline{A(B + C) + D} = \overline{A(B + C)}.\overline{D} = [\overline{A} + \overline{(B + C)}]\overline{D} = (\overline{A} + \overline{B}.\overline{C})\overline{D}$

In this section we will learn how to use Boolean algebra and DeMorgan's theorems to simplify Boolean expressions.

**Example 1:**
Prove that $\overline{A.B}.B = \overline{A}.B$

$$
\begin{aligned}
&\overline{A.B}.B \\
&= (\overline{A} + \overline{B})B \\
&= \overline{A}.B + \overline{B}.B \\
&= \overline{A}.B + 0 \\
&= \overline{A}.B
\end{aligned}
$$

**Example 2:**
Prove that $\overline{A.B}.\overline{B + C} = \overline{B}.\overline{C}$

$$
\begin{aligned}
&\overline{A.B}.\overline{B + C} \\
&= (\overline{A} + \overline{B})\overline{B}.\overline{C} \\
&= \overline{A}.\overline{B}.\overline{C} + \overline{B}.\overline{B}.\overline{C} \\
&= \overline{A}.\overline{B}.\overline{C} + \overline{B}.\overline{C} \\
&= \overline{B}.\overline{C}(A + 1) \\
&= \overline{B}.\overline{C}
\end{aligned}
$$

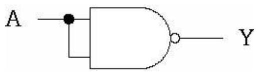
**Example 3:**
Prove that $\overline{A.B(\overline{C + D}).AB} = \overline{A} + \overline{B} + C + D$

$$
\begin{aligned}
&\overline{A.B(\overline{C + D}).AB} \\
&= \overline{AB(\overline{C + D})} + \overline{AB} \quad \textit{Break the outer most bar} \\
&= \overline{AB} + \overline{\overline{C + D}} + \overline{A} + \overline{B} \\
&= \overline{A} + \overline{B} + C + D + \overline{A} + \overline{B} \\
&= \overline{A} + \overline{B} + C + D
\end{aligned}
$$

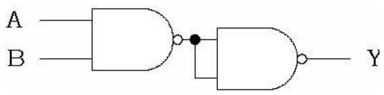
## 3.6 Universality of NAND and NOR gates

All Boolean expressions can be implemented using the fundamental AND, OR and NOT gates. It is possible, however, to implement any Boolean expressions using only NAND gates. This is because NAND gate can be used to make other types of gates AND, OR and NOT. For that reason, NAND gate is known as a universal gate. The circuits below shows the construction of the NOT, AND and OR gates utilizing only NAND gates.

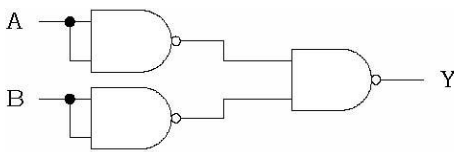
The expression $Y = \overline{A.A} = \overline{A}$ is equivalent to a NOT gate.



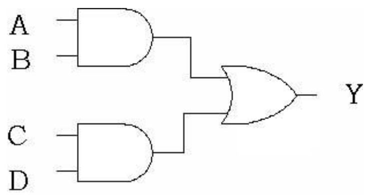

The expression $Y = \overline{\overline{AB}} = AB$ is equivalent to an AND gate.




The expression $Y = \overline{\overline{A}.\overline{B}} = \overline{\overline{A}} + \overline{\overline{B}} = A + B$ is equivalent to an OR gate.




The SOP expression can be constructed by using only NAND gates. Let us construct the expression $Y = AB + CD$ using the basic logic gates. This requires two ICs, a two input AND gate (7408) and a two input OR gate (7432).




This expression can be accomplished by using only NAND gate which requires only one NAND gate IC (7400).

1. Double invert the expression $Y = AB + CD$

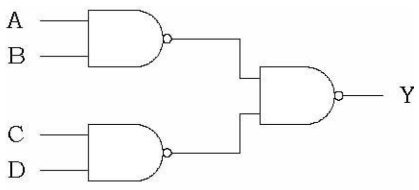
$$Y = \overline{\overline{AB + CD}}$$

2. Keep the top inversion bar.

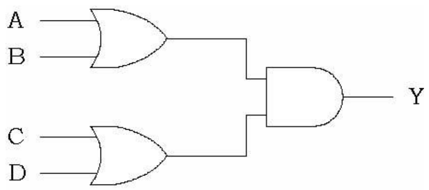
3. Apply DeMorgan's theorem $\left(\overline{A+B}=\overline{A}.\overline{B}\right)$ to the bottom inversion bar to eliminate the OR operation.

   $Y=\overline{\overline{AB}.\overline{CD}}$

4. Draw the circuit using only NAND gates.




The POS expression can be constructed by using only NOR gates. Let us construct the expression $Y=(A+B)(C+D)$ using the basic logic gates. This requires two ICs, a two input OR gate (7432) and a two input AND gate (7408).




This expression can be accomplished by using only NOR gate which requires only one NOR gate IC (7402).

1. Double invert the expression $Y=(A+B)(C+D)$

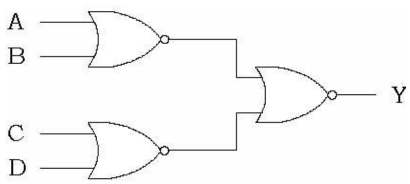
   $Y=\overline{\overline{(A+B)(C+D)}}$

2. Keep the top inversion bar.

3. Apply DeMorgan's theorem $(\overline{A.B}=\overline{A}+\overline{B})$ to the bottom inversion bar to eliminate the AND operation.

   $Y=\overline{\left(\overline{A+B}\right)+\left(\overline{C+D}\right)}$

4. Draw the circuit using only NOR gates.



## 3.7 Exercises

1. Write the SOP and POS expressions:

   a. 2-input AND gate
   b. 2-input NOR gate
   c. 2-input NAND gate
   d. 2-input XOR gate
   e. 2-input XNOR gate

2. Write the SOP and POS expressions:

   a. 3-input OR gate
   b. 3-input XOR gate
   c. 3-input XNOR gate.

3. Prove that the SOP and POS expressions are equivalent.

   a. 2-input AND gate
   b. 2-input NOR gate
   c. 2-input NAND gate
   d. 2-input XOR gate
   e. 2-input XNOR gate

4. Prove the expressions using Boolean algebra.

   a. $(A+\overline{B})(\overline{A}+B)(\overline{A}+\overline{B}) = \overline{A}.\overline{B}$
   b. $(A+B)(\overline{A}+\overline{B}) = \overline{A}B + A\overline{B}$
   c. $\overline{A}.\overline{B} + \overline{A}B + A\overline{B} = \overline{A} + \overline{B}$
   d. $(A+B)(A+C) = A + BC$
   e. $\overline{A}B\overline{C} + AB\overline{C} + B\overline{C}D = B\overline{C}$
   f. $AB + A\overline{B} + CB + C\overline{B} = A + C$
   g. $AB + A\overline{B} + \overline{A}.\overline{B} + \overline{A}C + \overline{A}.\overline{C} = 1$
   h. $(A+B)BC + A = BC + A$
   i. $(A+B)\overline{B} + \overline{B} + BC = \overline{B} + C$
   j. $(A+\overline{B})(B+C)B = AB$
   k. $ABCD + AB\overline{C}D + ABC\overline{D} + A.\overline{B}.\overline{C}.D + AB\overline{C}.\overline{D} + A\overline{B}.\overline{C}.\overline{D} + A\overline{B}.C = A$
   l. $ABC + BC + \overline{B}C = C$
   m. $\overline{A}B\overline{C} + A\overline{B}C + AB\overline{C} + AB + A\overline{C} = A + B\overline{C}$

5. Prove the expressions using Boolean algebra and DeMorgan's theorems.

   a. $\overline{\overline{A}.\overline{B}.(A+B)} = 1$
   b. $\overline{A.\overline{B}} + \overline{A.(\overline{A}+C)} = \overline{A} + B + \overline{C}$

c. $\overline{A\overline{B}.(A+C)} + \overline{A}.B.\overline{(A+B+C)} = \overline{A}+B$

d. $\overline{\overline{\overline{\overline{A}B}C}D} = (A+\overline{B})C + \overline{D}$

e. $\overline{(A+\overline{B})(\overline{A}+B)} = \overline{A}B + A\overline{B}$

f. $\overline{B}(A+\overline{C}) + \overline{\overline{A}BC + \overline{C(\overline{A}+B)}} = A+\overline{B}$

g. $\overline{(\overline{A}+B+C+D)} + A.\overline{B}.\overline{C}.D = A.\overline{B}.\overline{C}$

h. $\overline{\overline{A}.\overline{B}.\overline{AB}} = A+B$

i. $\overline{\overline{(AB+C)}(\overline{A}+\overline{B})\overline{C}} = AB+C$

j. $\overline{\overline{\overline{A.\overline{B}.\overline{C}}+BC}(\overline{AB})} = \overline{A}+B+C$

k. $\overline{\overline{\overline{(A+B)}(C+D)}(A+C)} = \overline{A}.\overline{B} + \overline{A}.\overline{C}$

6. Draw the expressions using only NAND gates.

a. $Y = AB + C$
b. $Y = ABC + D$
c. $Y = \overline{A}.\overline{B} + \overline{C}.D$
d. $Y = \overline{A}.\overline{B} + \overline{A}.B + A.B$
e. $Y = (A+B)(C+D)$

7. Draw the expressions using only NOR gates.

a. $Y = (A+B)C$
b. $Y = (A+B)(\overline{C}+\overline{D})$
c. $Y = A + BC$
d. $Y = A + BCD$
e. $Y = AB + C + D$